2021-09

APPLYING CONVOLUTIONAL NEURAL NETWORKS TO IDENTIFY MOVING TARGETS IN SAR IMAGERY

Henegar, Erik L.

Monterey, CA; Naval Postgraduate School

NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

APPLYING CONVOLUTIONAL NEURAL NETWORKS TO IDENTIFY MOVING TARGETS IN SAR IMAGERY

by

Erik L. Henegar

September 2021

Thesis Advisor: David A. Garren
Second Reader: David C. Jenn

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			*Form Approved OMB No. 0704-0188*

1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** September 2021	**3. REPORT TYPE AND DATES COVERED** Master's thesis

4. TITLE AND SUBTITLE
APPLYING CONVOLUTIONAL NEURAL NETWORKS TO IDENTIFY MOVING TARGETS IN SAR IMAGERY

5. FUNDING NUMBERS

6. AUTHOR(S) Erik L. Henegar

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey, CA 93943-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
N/A

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release. Distribution is unlimited.

12b. DISTRIBUTION CODE
A

13. ABSTRACT (maximum 200 words)

Synthetic Aperture Radar (SAR) is a type of radar that can provide high resolution imagery regardless of time of day or weather conditions. Convolutional Neural Networks (CNNs) or other deep learning algorithms can be applied to SAR imagery to conduct Automatic Target Recognition (ATR) of high value targets. SAR is a valuable reconnaissance and surveillance capability, but it is limited in its ability to show moving targets. In SAR imagery, moving targets appear smeared, making it difficult to perform ATR. This thesis analyzed various methods for performing ATR of moving targets in SAR imagery using CNNs. Analysis was conducted through computer simulation using the Synthetic and Measured Paired and Labeled Experiment (SAMPLE) dataset to train and test the classification accuracy of a CNN algorithm. This thesis determined that out of the various analyzed methods for classifying moving targets using a CNN, the most accurate classification occurred when the CNN was trained using images of moving targets. Autofocus image processing techniques were shown to improve classification accuracy but not to acceptable levels. Future research is recommended to improve autofocus image processing techniques and to develop a method to separate stationary and moving target images for classification by CNNs trained on stationary or moving target data.

14. SUBJECT TERMS synthetic aperture radar, SAR, convolutional neural network, CNN, deep learning, DL, automatic target recognition, ATR	**15. NUMBER OF PAGES** 69
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	**18. SECURITY CLASSIFICATION OF THIS PAGE**	**19. SECURITY CLASSIFICATION OF ABSTRACT**	**20. LIMITATION OF ABSTRACT**
Unclassified	Unclassified	Unclassified	UU

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release. Distribution is unlimited.

APPLYING CONVOLUTIONAL NEURAL NETWORKS TO IDENTIFY MOVING TARGETS IN SAR IMAGERY

Erik L. Henegar
Captain, United States Marine Corps
BS, United States Naval Academy, 2015

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
September 2021

Approved by: David A. Garren
Advisor

David C. Jenn
Second Reader

Douglas J. Fouts
Chair, Department of Electrical and Computer Engineering

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Synthetic Aperture Radar (SAR) is a type of radar that can provide high resolution imagery regardless of time of day or weather conditions. Convolutional Neural Networks (CNNs) or other deep learning algorithms can be applied to SAR imagery to conduct Automatic Target Recognition (ATR) of high value targets. SAR is a valuable reconnaissance and surveillance capability, but it is limited in its ability to show moving targets. In SAR imagery, moving targets appear smeared, making it difficult to perform ATR. This thesis analyzed various methods for performing ATR of moving targets in SAR imagery using CNNs. Analysis was conducted through computer simulation using the Synthetic and Measured Paired and Labeled Experiment (SAMPLE) dataset to train and test the classification accuracy of a CNN algorithm. This thesis determined that out of the various analyzed methods for classifying moving targets using a CNN, the most accurate classification occurred when the CNN was trained using images of moving targets. Autofocus image processing techniques were shown to improve classification accuracy but not to acceptable levels. Future research is recommended to improve autofocus image processing techniques and to develop a method to separate stationary and moving target images for classification by CNNs trained on stationary or moving target data.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS AND ABBREVIATIONS

ATR	Automatic Target Recognition
CNN	Convolutional Neural Network
DL	Deep Learning
EO	Electro-Optical
ISAR	Inverse Synthetic Aperture Radar
ML	Machine Learning
MSTAR	Moving and Stationary Target Acquisition and Recognition
PGA	Phase Gradient Autofocus
RAR	Real Aperture Radar
ReLu	Rectified Linear Unit
SAMPLE	Synthetic and Measured Paired Labeled Experiment
SAR	Synthetic Aperture Radar
SLAR	Side-Looking Aperture Radar

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

I would like to thank my professors without whom I could not have completed my course of study while at the Naval Postgraduate School. My thanks especially to my thesis advisor, Professor David Garren, and my second reader, Professor David Jenn, for their assistance and guidance throughout the thesis process.

Most importantly, I would like to thank my wife, Amanda, who gave birth to our first child during this time. Thank you for being so strong, patient, and understanding. You are an incredible wife and mother, and I would not have been able to do any of this without you.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

The analysis and classification of military targets within imagery captured by aerial or spaceborne sensors provides the intelligence community with valuable information about enemy force size, disposition, and possible intentions. High resolution satellite imagery collected by electro-optical (EO) sensors is used in everyday life on navigation applications that the average person has access to on their cellular phone. In this type of imagery, it is easy for a person to visually identify a wide variety of objects to include military targets. EO sensors are unfortunately limited and unable to collect imagery in low light and poor weather conditions. In a military setting, waiting for daylight or clear skies to collect imagery could delay critical decision making or altogether prevent friendly forces from having information on adversary troop and equipment movements. Synthetic aperture radar (SAR) is a sensor modality that can bridge the gap left by the limitations of EO sensors. SAR sensors combine radar hardware and relative motion to generate photograph-like imagery of targets and areas of interest [1]. Unlike EO sensors, SAR is able to generate imagery in any illumination and almost any weather condition [2]. SAR operated at low center frequencies can even penetrate cover and concealment such as foliage and bunkers [1].

SAR imagery does have limitations. SAR imagery is produced in grayscale and does not have quite as high resolution as EO imagery. SAR sensors generally do not produce clear imagery of moving targets. SAR sensors work in a similar manner to a camera with a long exposure time, causing moving objects to look blurred or smeared in SAR imagery. These attributes of SAR imagery along with the vast amount of imagery generally collected make it a difficult task for humans to accurately identify targets. Machine learning (ML) provides an ideal solution to this problem.

Automatic target recognition (ATR) is a technique that uses machine learning to automatically identify targets within SAR imagery. ML can easily screen large amounts of imagery and accurately identify targets. Deep learning (DL) is a machine learning method used in ATR that requires little image processing and has high recognition accuracy. Within DL, convolutional neural network (CNN) algorithms have presented especially outstanding performance. ML, including CNNs, requires large amounts of training imagery

on the desired targets to be identified to be able to recognize those targets. ML used for the purpose of ATR is not a new concept, but the use of machine learning for the identification of moving targets appears to be a relatively new topic of endeavor.

A. PURPOSE

The objective of this thesis is to develop a CNN that accurately identifies multiple types of stationary and moving targets within SAR imagery. To confirm the accuracy of the CNN, target images were taken from the Synthetic and Measured Paired Labeled Experiment (SAMPLE) dataset [3]. The SAMPLE dataset was created by the Air Force Research Laboratory to augment and improve upon the Moving and Stationary Target Acquisition and Recognition (MSTAR) dataset [2]. The MSTAR dataset contains measured data whereas, the SAMPLE dataset contains synthetically generated data [2]. Using the synthetic data forces the algorithm to train and learn how to identify targets regardless of the environment where the target exists [2]. The SAMPLE dataset was also used to test the CNN recognition accuracy on moving targets. The MATLAB deep learning toolbox was used for simulation and analysis during this study [4].

B. ORGANIZATION

Chapter II of this thesis provides a more in-depth description of SAR data collection, CNN structure and application, and how SAR and CNNs work together in an ATR application. Chapter III delves into the design of the CNN, the datasets used, as well as the parameters of the simulations used to test the performance of the CNN. Chapter IV provides a detailed analysis of the simulation results. A summary of the results, conclusion, and recommended future work are presented in Chapter V.

II. TECHNICAL BACKGROUND

A. SYNTHETIC APERTURE RADAR

Radar is a remote sensing system that is used for many purposes including detection, tracking, and imaging. Until the early 1950s, real aperture radars performed limited imaging tasks, but they had poor resolution [3] in the radar cross-range directions. Initially, many RAR systems would rotate the antenna to scan the target area, but in the early 1950s, engineers realized that they could instead fix the radar on the fuselage of an aircraft to increase aperture length and improve along-track resolution [3]. This new radar imaging method was called side-looking aperture radar (SLAR) [5]. In 1951, Carl Wiley of the Goodyear Aircraft Corporation developed the technology known today as SAR [3]. Wiley developed a method to analyze Doppler frequency as it applied to a coherent, moving radar [3]. This Doppler frequency analysis allowed for the azimuth resolution of the received signal to be enhanced by separating the returns into groups based on the Doppler shift of each return [5]. SAR is a real aperture radar system mounted on a moving vehicle, generally an air or space craft. The movement of the real aperture radar along the axis of motion emulates a much longer aperture [4]. The SAR system tracks the movement of the platform it is mounted on and uses this information in conjunction with the time of signal transmission and reception to construct the synthetic aperture that is much longer than the length of the physical antenna [6].

The derivation that follows is summarized from [5]. For RAR systems, with signal wavelength λ and antenna length L, the beamwidth θ_B is equal to

$$\theta_B = \frac{\lambda}{L} \tag{2.1}$$

For a given range R, the cross-range resolution is

$$\delta_{cr} = R\theta_B \tag{2.2}$$

If the radar is installed on an aircraft or satellite flying at speed v, the time on target or aperture time T_a is equal to

$$T_a = \frac{\theta_B}{v / R} \tag{2.3}$$

For the time on target, the airborne vehicle will fly distance

$$d = vT_a = R\theta_B \tag{2.4}$$

If the airborne radar system is a SAR system, the distance flown will be equal to the length of the synthetic antenna

$$L_{SAR} = d = R\theta_B \tag{2.5}$$

The SAR beamwidth is

$$\theta_{SAR} = \frac{\lambda}{2L_{SAR}} \tag{2.6}$$

The factor 2 in the denominator of the above equation is due to the fact that the phase shift in signal between elements in SAR is twice the phase shift in real arrays because the pulses in SAR are transmitted sequentially by the array elements. For the same distance R as before, the cross-range resolution for SAR is

$$\delta_{SAR} = R * \theta_{SAR} = R\frac{\lambda}{2L_{SAR}} = R\frac{\lambda}{2R\theta_B} = \frac{L}{2} \tag{2.7}$$

1. SAR Operating Modes

There are three main SAR imaging modes. These modes are Stripmap, Scan SAR, and Spotlight [6]. These modes each have their own strengths and weaknesses depending on the size and type of target to be imaged.

a. Stripmap SAR

Stripmap mode is a fundamental operating mode for SAR. The SAR radar images one swath of a continuous, single strip [4]. Swath width for an airborne radar can range

from a few kilometers to 20 km, and for spaceborne radars the swath width can be 30 to 500 km [6]. The stripmap SAR scanning mode is shown in Figure 1.

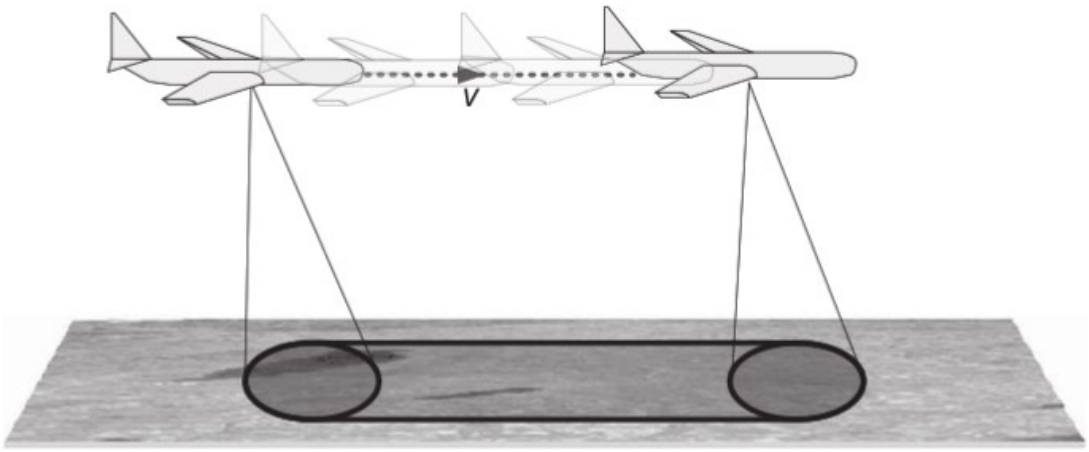

Figure 1. Stripmap SAR scanning mode. Source: [7].

b. Scan SAR

Scan SAR is used when the desired area to be imaged requires a swath wider than the ambiguous range [4]. The radar steers the antenna elevation pattern to different elevation angles corresponding to multiple sub-swaths [4]. This operating mode produces imaging for a wide swath with poor azimuth resolution [6]. Figure 2 shows a visual representation of the scan SAR mode.

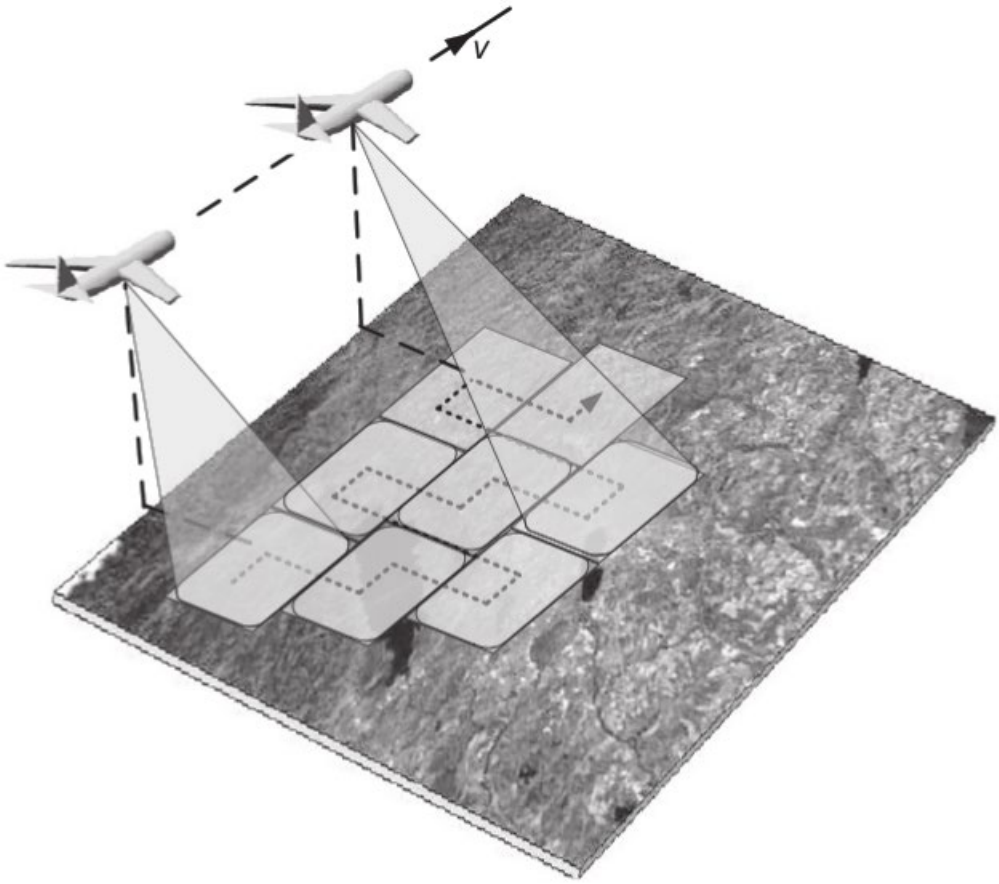

Figure 2. Scan SAR scanning mode. Source: [7].

c. Spotlight SAR

Spotlight SAR steers the antenna pattern towards a fixed point. The longer illumination time on the desired target results in increased synthetic aperture length and better resolution [6]. Spotlight SAR is diagrammed in Figure 3.

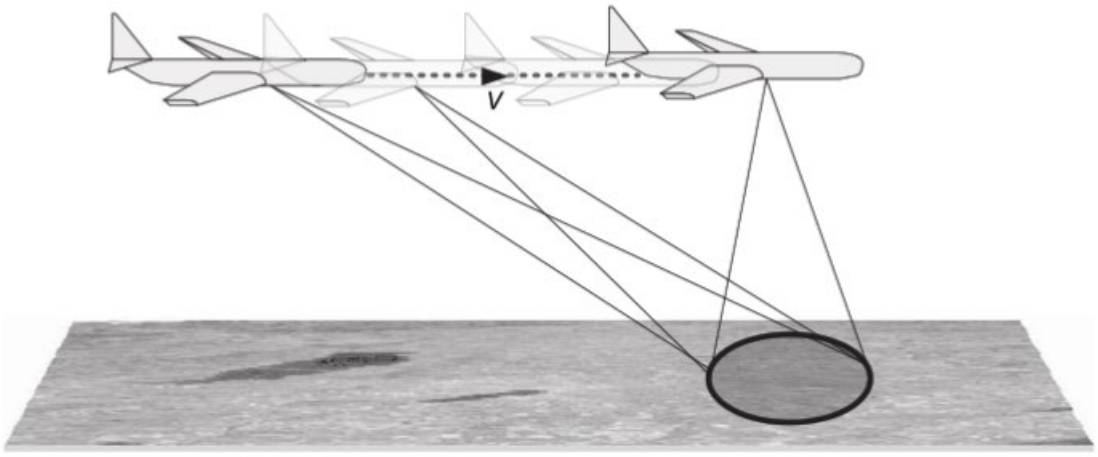

Figure 3. Spotlight SAR scanning mode. Source: [7].

d. Inverse SAR

Inverse SAR (ISAR) is a variation of SAR where the targets are in motion, and the radar is stationary relative to the motion of the targets. As the target moves, the look angle of the target changes with respect to the radar line of sight axis [6]. The change in angle in the data received by the ISAR is used to resolve the different points on the cross range axis [7]. ISAR cannot be used to image stationary objects, which means ISAR is not ideal for use in operating spaces where both stationary and moving targets exist. ISAR is illustrated in Figure 4.

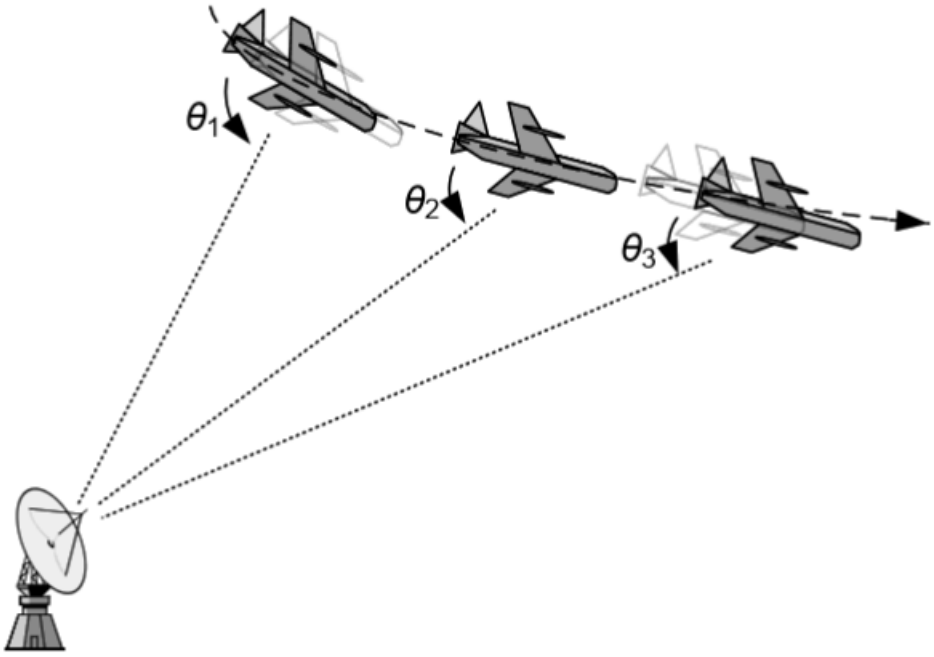

Figure 4. ISAR scanning mode. Source: [7].

2. Phase Errors in SAR Imagery

As previously discussed, Doppler frequency analysis is the key technique that allows for the transformation of received radar data into SAR imagery. For the SAR radar system to properly conduct the Doppler frequency analysis, the motion and rate of motion of the radar platform must be known. Any error in the measured motion of the radar platform causes phase errors in the imagery output of the SAR system.

a. Types of Phase Errors

There are two types of phase errors that occur in SAR imagery. The first and most common type is quadratic phase error, and the second type is high-frequency phase error [7]. Quadratic phase errors occur as a result of inaccurate knowledge of the SAR platform position usually caused by errors in measuring the platform velocity [7]. Quadratic phase errors cause blurring in SAR imagery [7]. These errors can also occur when the platform velocity is measured correctly, but objects within the radar's field of view are not stationary. The movement of objects within the radar's field of view changes the motion of the radar platform in relation to the moving object, resulting in the same error as an

incorrect velocity measurement. This type of phase error is explained mathematically by the following equations [8].

The demodulation time τ_0 for SAR is equal to

$$\tau_0 = \frac{2}{c}\sqrt{R_0^2 + d^2} \approx \frac{1}{c}\left(2R_0 + \frac{d^2}{R_0}\right) = \frac{1}{c}\left(2R_0 + \frac{(Vt)^2}{R_0}\right) \tag{2.8}$$

where c is the speed of light, R_0 is the distance from the radar to the scene center, d is distance from the radar to the center of the aperture, V is the measured platform velocity, and t is time.

If the actual platform velocity is V_a, then demodulation error becomes

$$\tau_0' \approx \frac{1}{c}\left(2R_0 + \frac{(V_a t)^2}{R_0}\right) \tag{2.9}$$

This difference in velocity leads to a demodulation timing error equal to

$$\varepsilon(t) = \tau_0 - \tau_0' = \frac{1}{cR_0}\left(V_a^2 - V^2\right)t^2 \tag{2.10}$$

To convert the timing error to a phase error, $\varepsilon(t)$ is multiplied by ω_0, which is the center frequency of the radar.

$$\phi(t) = \varepsilon(t)\omega_0 = \frac{\omega_0}{cR_0}\left(V_a^2 - V^2\right)t^2 \tag{2.11}$$

When the phase error is due to along-the-line-of-sight position errors, $\delta R(t)$, such as an object moving in the field of view, the phase error equation can be represented as

$$\phi(t) = \frac{2\delta R(t)}{c}\omega_0 = \frac{4\pi}{\lambda}\delta R(t) \tag{2.12}$$

As previously mentioned, the second type of SAR phase error is a high-frequency error. High-frequency errors are caused by signal propagation effects or by uncompensated platform motion errors. These errors result in a loss of contrast in SAR images [8].

b. Correction of Phase Errors

There are three methods for correcting SAR image phase errors. The two traditional methods are inverse filtering and map-drift, and the more modern technique is phase gradient autofocus (PGA) [7]. The inverse filtering technique attempts to isolate a single point target within the SAR image [7]. Once the point target is isolated, a window is applied around the point to focus on the data immediately surrounding the point and ignore all image data outside the window [7]. The image data within the window is then azimuthally decompressed using a one-dimensional Fourier transformation and the phase error is measured [7]. The SAR image is then corrected by multiplying the total image data by the complex conjugate of the phase error function and applying azimuth compression [7]. Inverse filtering can be an effective technique, but it is challenging to find a truly isolated point target in most SAR imagery [7]. If the point target is not totally isolated, there will be residual streaking in the corrected SAR image [8].

The second traditional phase correction method is the map-drift technique. The map drift technique analyzes all of the SAR image data and assumes that the phase error can be estimated as a quadratic polynomial [7]. The map drift technique is a more robust technique than the inverse filtering technique, but it loses reliability when the phase error function cannot be adequately described as a low order polynomial [8].

The third and more modern method of phase error correction is PGA. PGA is a multi-step, non-parametric process [7]. The first step in the PGA process is similar to the first step of the inverse filtering process [7]. The PGA selects one isolated point target per range line within the SAR image and circularly shifts each target point to the center of the scene [7]. This shifting essentially creates a new image with all the point targets stacked in the scene center [7]. In the next step, the PGA applies a rectangular window around the points that were center shifted [7]. The data outside of this window is considered noise and is ignored for the purpose of phase estimation [7]. The targets within the window are then

decompressed using a one-dimensional discrete Fourier transform [7]. A maximum-likelihood estimation of the phase error is made for all data points within the established window [7]. PGA then corrects the image in the same manner as the inverse filtering technique [7]. Sometimes PGA is executed in an iterative manner with a smaller window being used in each iteration and resulting in better phase error correction performance [8]. The PGA algorithm process is illustrated in Figure 5.

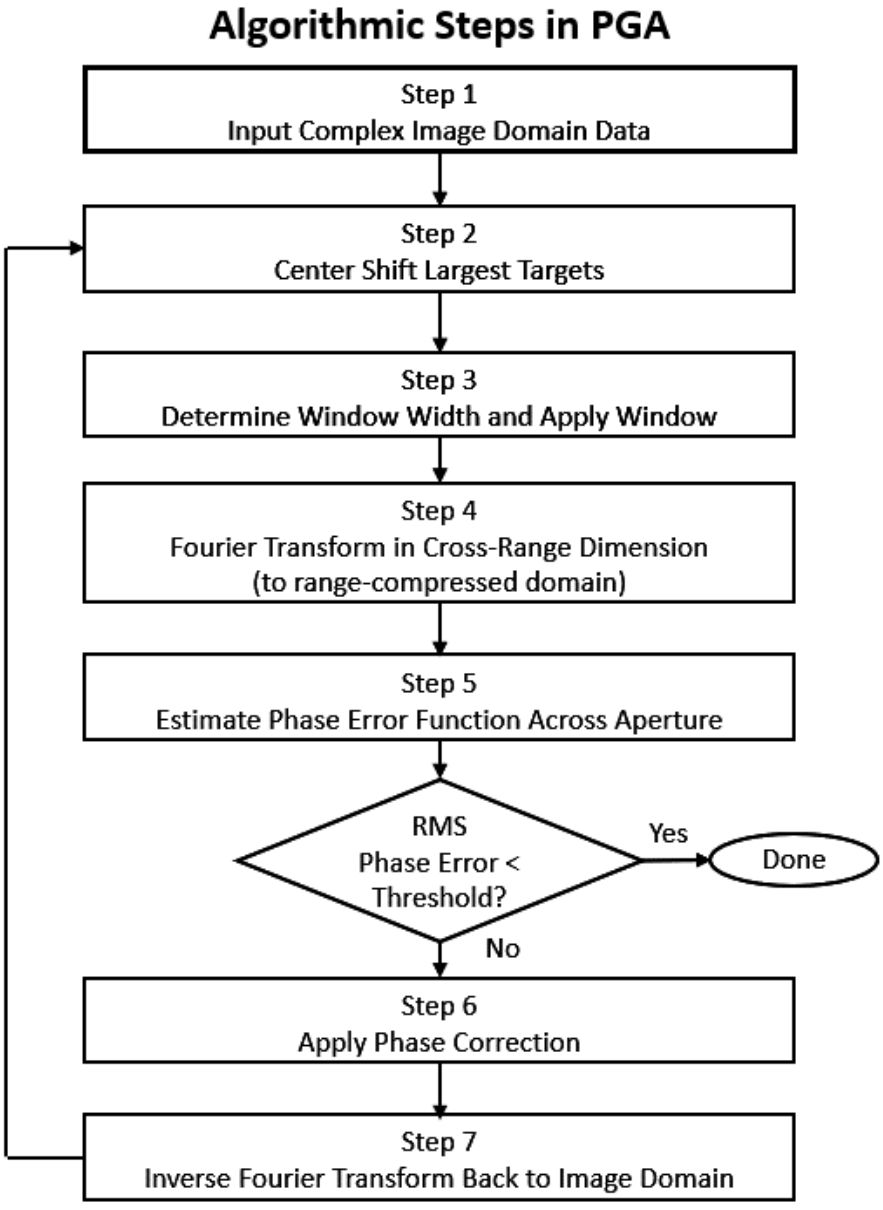

Figure 5. Flow diagram of the PGA algorithm. Source: [8].

B. CONVOLUTIONAL NEURAL NETWORK

Convolutional neural networks are a type of deep learning method designed to process data and automatically and adaptively learn features and patterns [9]. Deep learning is a form of machine learning with multiple layers of neurons that perform coarse to fine analysis of data for decision making [10]. CNNs became a dominant deep learning method used in image processing in 2012 after the ImageNet Large Scale Visual Recognition Competition [9]. At the competition, a CNN proved to have much better accuracy results than other methods of machine learning.

1. CNN Layers

CNNs use multiple layers to learn and identify features in images. There are three basic layers used in CNNs. Theses layers are convolution layers, pooling layers, and fully connected layers [10]. Each layer type has a different purpose in enabling the CNN to learn and classify targets in images.

a. Data Input

The data input for a CNN is a tensor with shape $T = w \times h \times d \times I$, where w is the image width in pixels, h is the image height in pixels, d is the image depth, and I is the number of images [9]. The image depth depends on the type of data that makes up the image. If the image data is purely magnitude, then $d = 1$, but if the image data is complex and includes phase, then $d = 2$ [10].

b. Convolution Layer

The convolution layer consists of a combination of linear and nonlinear operations [8]. The linear operations are convolution, and the nonlinear operations are activation functions [8]. Convolution is used for image feature extraction [8]. In convolution, a small array of numbers called a kernel is multiplied element-wise by a portion of the input tensor array that matches the array size of the kernel [8]. The elements of the product array are then summed to obtain the output tensor value for the corresponding position [8]. This process is repeated until the entire input tensor has been multiplied by the kernel for feature

extraction [8]. Kernels are usually 3 × 3, but they can be 5 × 5 or 7 × 7 [9]. The CNN convolution operation is displayed in Figure 6.

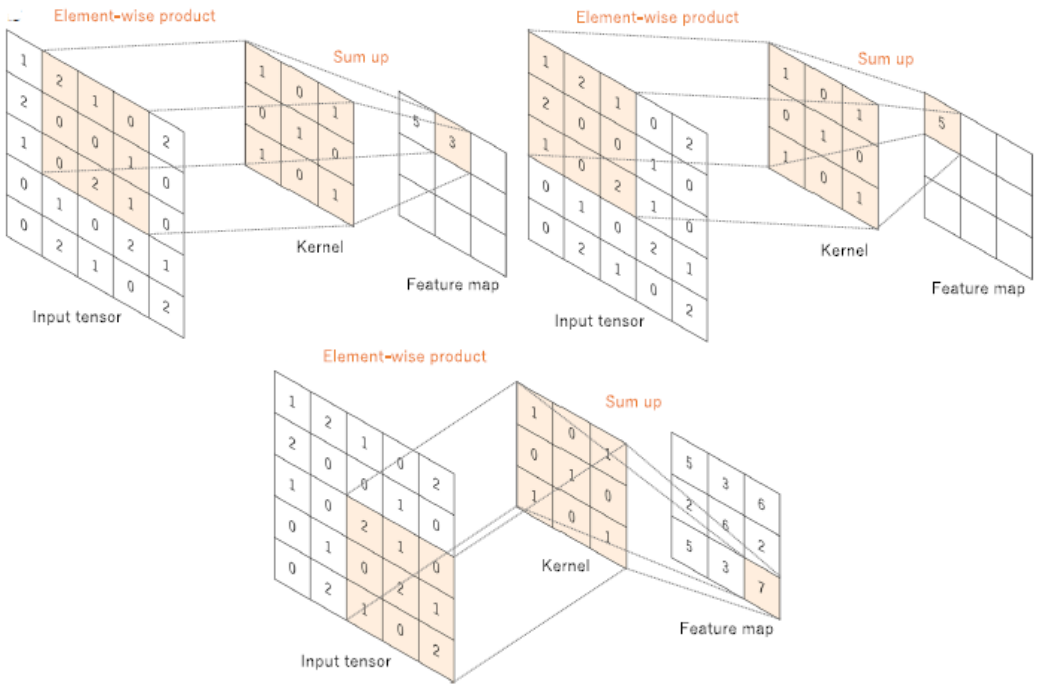

Figure 6. Convolution operation with 3 × 3 kernel size, no zero padding, and stride of 1. Adapted from [9].

The output of the convolution layer is optimized using three hyperparameters. These hyperparameters are depth, stride, and zero padding [9]. Depth refers to the number of neurons in a layer that connect to the same region in the input tensor [9]. Stride controls the number of units the kernel filter translates at a time per output. For example, if the stride is one, the kernel filter will shift by one pixel per output on the feature map [10]. Zero padding solves two problems that arise during convolution. The first problem is that the center of each kernel cannot overlap with the outermost elements of the input tensor [8]. The second problem is that the resulting output feature map is a smaller array than the input tensor [8]. Zero padding involves adding rows and columns of zeros to the outside of the input tensor [9].

The nonlinear activation function is the second portion of the convolution layer. The most commonly used activation function is the rectified linear unit (ReLU). ReLU computes the function $f(x) = \max(0, x)$ for each element of the feature map [9]. This function converts any negative value within the feature map to zero.

c. Pooling Layer

The pooling layer is a downsampling operation that is used to decrease filter sensitivity to small shifts and distortions, which also decreases the number of learnable parameters [8]. There are two different pooling operations, max pooling and global average pooling [9].

Max pooling extracts 2 × 2 sections from input tensors and outputs the maximum value from each section [8]. The pooling filters are two dimensional and are applied to every layer along the depth of the input [8]. This means that the input height and width are downsampled, but not the depth [8]. Max pooling is the most commonly used pooling layer operation [9]. The max pooling operation is demonstrated in Figure 7.

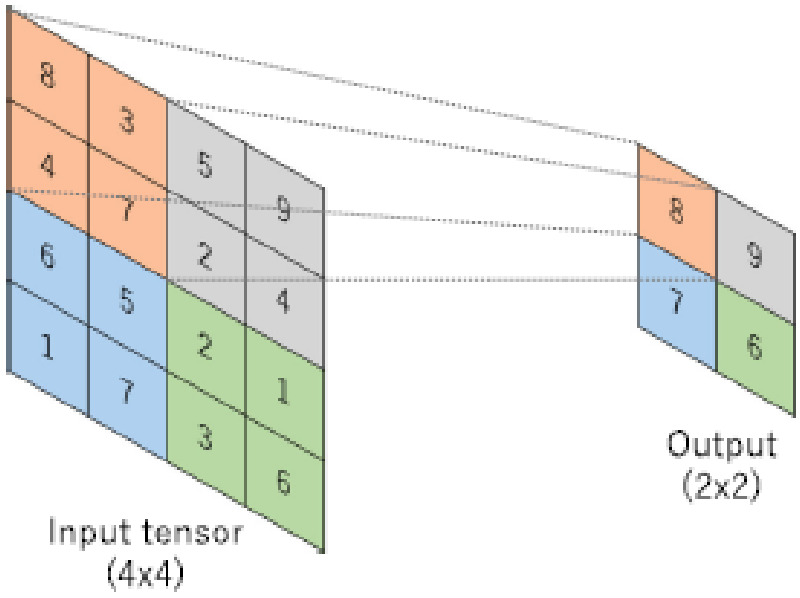

Figure 7. Max pooling operation with 2 × 2 filter size, no zero padding, and stride of 2. Source: [9].

Global average pooling is another pooling operation that takes the average of all elements in an input tensor resulting in a 1 × 1 output array [8]. This is an extreme form of downsampling that reduces the amount of learnable parameters and allows the CNN to accept multiple input sizes [9].

d. Fully Connected Layer

In the fully connected layer, the output feature maps from the final convolution or pooling layer are transformed into vectors and connected to one or more other fully connected layers [8]. Each input neuron is connected to an output neuron [8]. These connections allow the features extracted by the previous layers to be mapped to the final output target classes of the network [8]. All fully connected layers are followed by an activation function such as ReLU [8]. The final fully connected layer is followed by an activation function that is usually different from the previous functions. A commonly used final activation function is a Softmax function [8]. The Softmax function is a classifier that normalizes the outputs from the final fully connected layer to target class probabilities in the value range of 0 to 1 [8]. The probabilities of all target classes sum to 1. The CNN then classifies the input target as the target class with the highest probability.

2. CNN Architecture Design

The basic structure for CNNs is a convolution layer followed by a pooling layer followed by a fully connected layer. All CNNs are a variation of this basic structure that include one or multiples of the three basic layer types. Generally, adding more layers leads to more accurate classification, but adding too many layers can increase the effects of overfitting [11]. Overfitting has a negative effect of the training of a CNN and will be discussed further in the following section. Another consideration when determining the CNN architecture is computer memory. More CNN layers require more memory space to store the outputs of each layer. An example CNN architecture is shown in Figure 8.

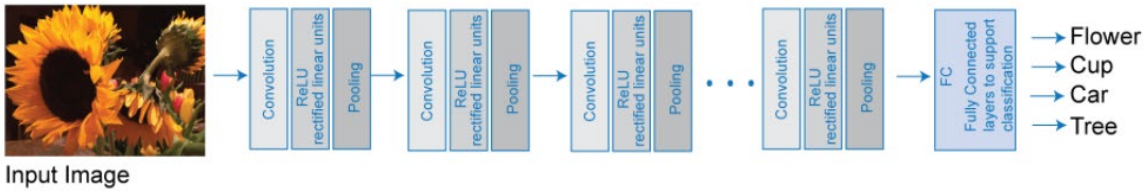

Figure 8. Example CNN architecture. Source: [12].

3. CNN Training

Training a CNN involves finding the values for kernels in the convolution layers and weights in the fully connected layers that minimize the differences between the network output predictions and the given target class labels for the training dataset [8]. A loss function and a gradient descent optimization algorithm play key roles in the CNN training process [8]. The loss function calculates performance under specific kernels and weights through forward propagation on the training dataset [8]. The performance is calculated by measuring the compatibility of output predictions and the given target class labels [8]. The values of the CNN kernels and weights are then updated according to the loss value through the back propagation optimization algorithm called gradient descent [8]. The kernels and weights are adjusted to minimize the difference between the output predictions and the target classes [8]. These adjustments are made using the negative direction of the gradient of the loss function and an arbitrary step size determined by the learning rate [8]. The gradient is formulated as [9]:

$$w = w - \alpha \times \frac{\partial L}{\partial w} \tag{2.13}$$

where w is the learnable parameter, α is the learning rate, and L is the loss function. A diagram of a CNN and the training process is shown in Figure 9.

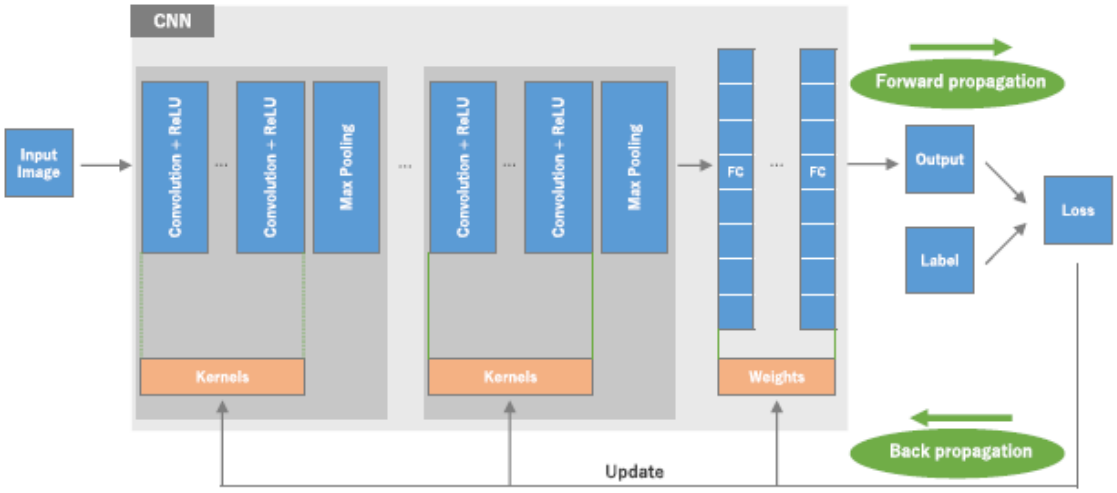

Figure 9. CNN structure showing forward propagation (loss function) and back propagation (gradient descent algorithm). Source: [9].

Training CNNs can sometimes result in a situation where the network learns regularities that are specific to the training dataset. This situation is called overfitting. Overfitting occurs when the CNN learns characteristics of the irrelevant noise in the input data rather than the characteristics of the target [9]. If the training dataset images are all taken in the same location, the CNN can learn characteristics relating to the environment around the target. The CNN will subsequently not be as accurate when identifying targets located in a different environment due to the difference in irrelevant noise.

THIS PAGE INTENTIONALLY LEFT BLANK

III. METHOD AND SIMULATION

The goal of this study is the recognition of military targets, both stationary and moving, using a CNN. The MATLAB deep learning toolbox is used to program, train, and test the CNN for this study.

A. SAMPLE DATASET

The Synthetic and Measured Paired Labeled Experiment (SAMPLE) dataset was chosen for training and testing of the CNN in this study. The SAMPLE dataset was developed at the Air Force Research Laboratory and is a combination of measured data and synthetic data for ten different military targets [2]. This dataset was developed to augment other measured data from SAR [2]. One of the measured datasets that SAMPLE is designed to augment is the Moving and Stationary Target Acquisition and Recognition (MSTAR) dataset [2]. The MSTAR dataset consists of one-foot resolution X-band (8-12 GHz) SAR images collected by various US government research organizations in the late 1990s [2]. The MSTAR data has been used for over two decades for the purpose of developing SAR ATR [2]. Table 1 shows the ten targets in the SAMPLE dataset and the comparison between the SAMPLE and MSTAR SAR images.

Table 1. Target Types and Comparison of SAMPLE and MSTAR Images. Adapted from [2].

Target	2S1	BMP2	BTR70	M1	M2	M35	M548	M60	T72	ZSU23
MSTAR										
SAMPLE										

One of the main reasons why the SAMPLE dataset was chosen over MSTAR for this study was to ensure that overfitting does not occur when training the CNN. The SAR imagery collected for the MSTAR dataset was collected in the same environment, meaning that CNNs being trained using this data could learn aspects of the background environment.

For the SAMPLE dataset, the synthetic image background noise is entirely random, meaning the CNN is not learning irrelevant background noise.

B. SIMULATION METHOD

The simulation for this study used 128 x 128-pixel SAMPLE dataset images as the input. These images were used to train and test the previously mentioned CNN using the MATLAB deep learning toolbox. The output of the CNN was a set of classification predictions that was characterized by a confusion matrix, which compared the predicted target labels to the actual target labels.

1. Simulation Input

The input for this simulation was 128 x 128-pixel SAR images from the SAMPLE dataset as previously stated. The SAR images in the SAMPLE dataset are stationary images of ten different targets. The SAR images of each target type are taken for target headings from 0° to 360° and at elevation angles from 14° to 18° [2]. Eight of the ten targets were used in the simulation training and testing to reduce the amount of data storage needed as well as the amount of time needed to train the CNN. To generate images of moving targets for this study, MATLAB code was used to generate artificial phase error vectors that were then applied to the SAMPLE SAR images. The phase error vectors are generated at random, so the error applied to each image is different. The phase error vectors when applied to the SAR images create the smearing effect that occurs when SAR receives returns from a moving target. Figures 10, 11, and 12, respectively, show a SAR target image without any phase errors, the phase errors applied to the SAR image, and the resulting SAR target image with phase errors. The color bars to the right of the SAR images are in units of dB and measure the intensity of the radar returns for each image pixel.

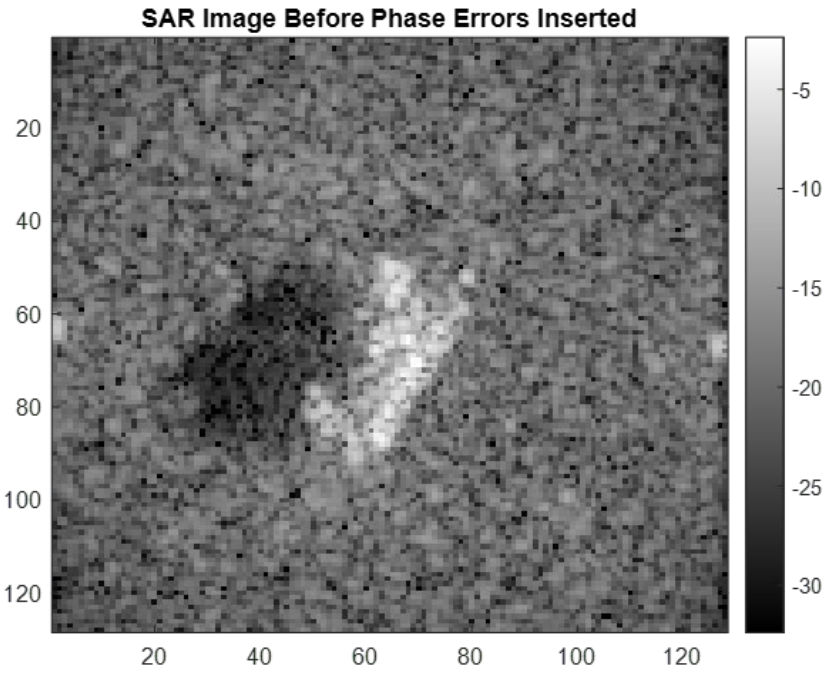

Figure 10. SAR stationary image prior to phase error insertion

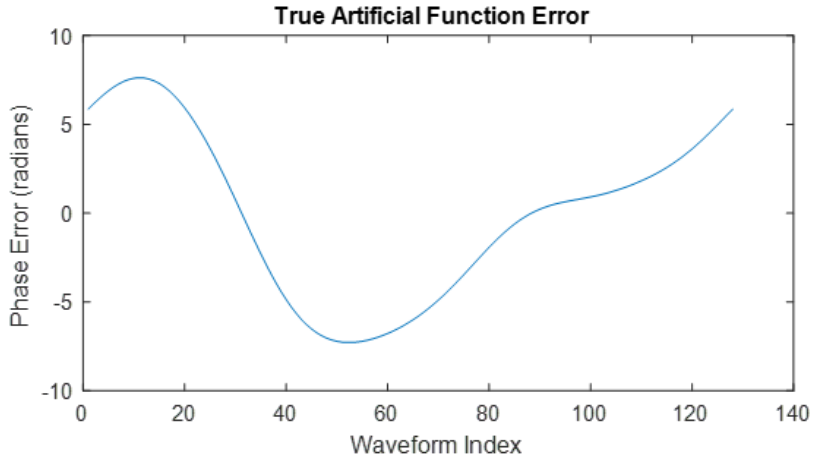

Figure 11. Phase error applied to stationary target to simulate moving target

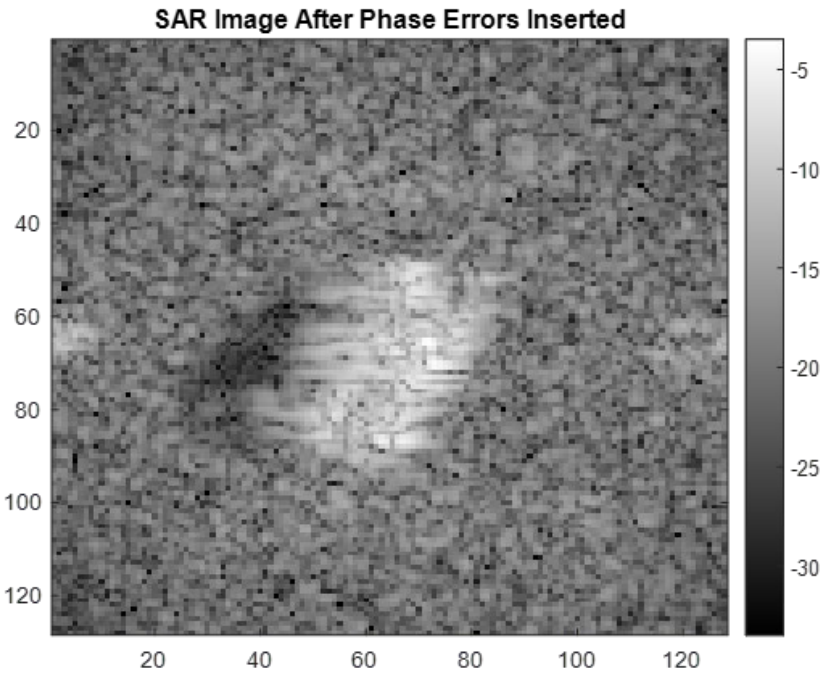

Figure 12. Simulated moving target SAR image

This study also looked at the using autofocusing as a method for improving CNN classification of moving targets. A PGA was applied to the moving target images generated within this study with the goal of estimating and correcting the phase errors applied to the stationary SAR images. Figure 13 shows a SAR target image without phase errors, Figure 14 shows the same target image but with applied phase errors, and Figure 15 shows the target image after autofocus has attempted to correct the applied phase errors.

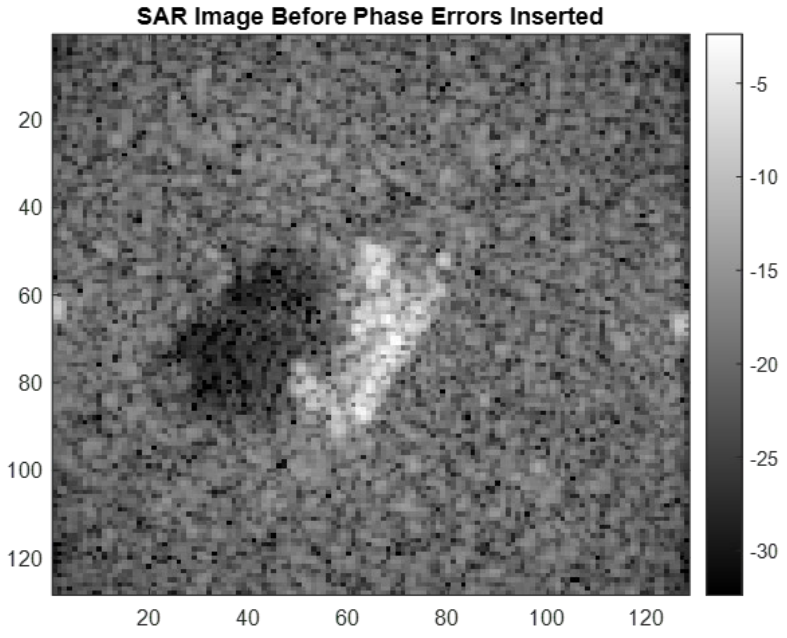

Figure 13. SAR image before phase error applied

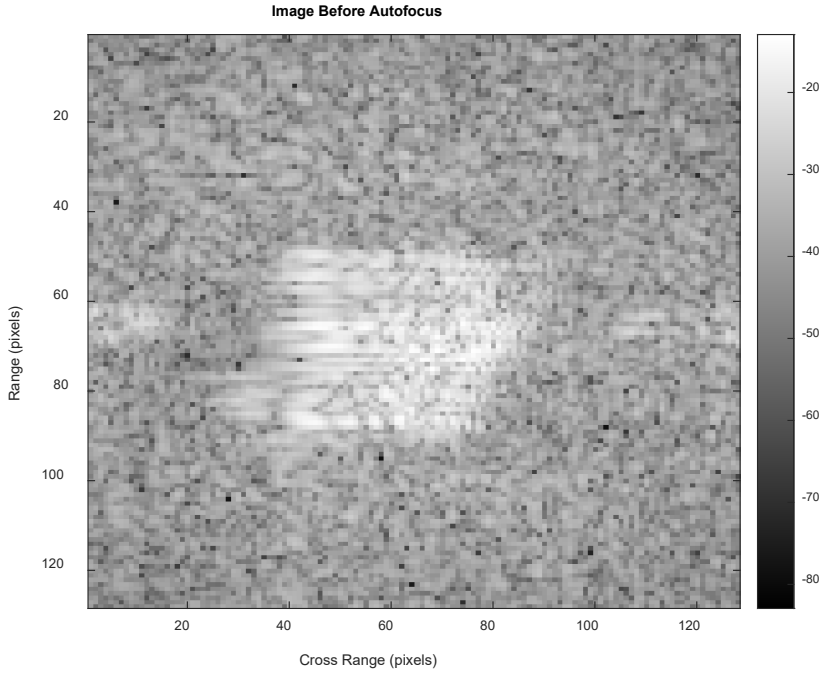

Figure 14. SAR image after phase errors applied to simulate moving target

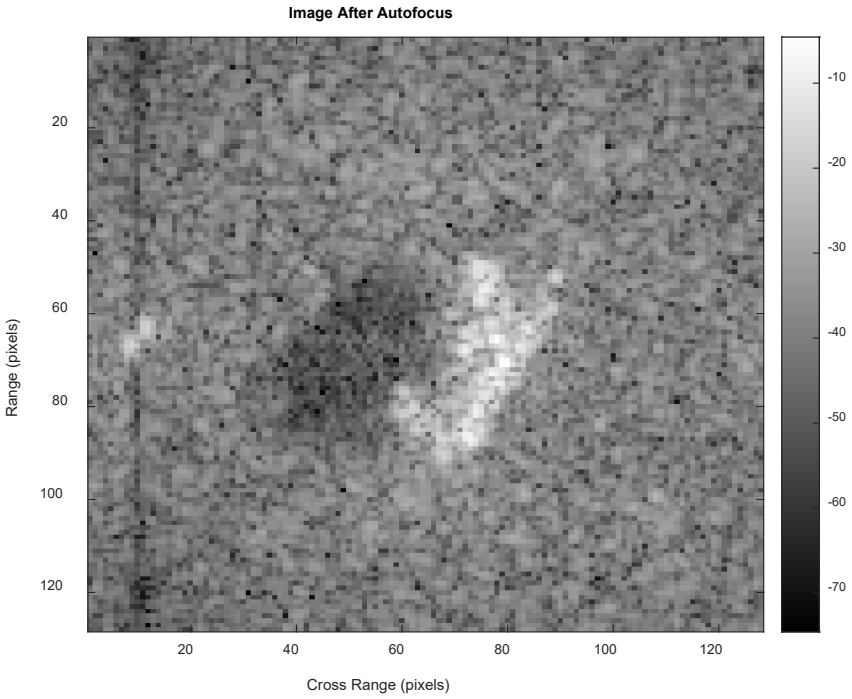

Figure 15. SAR image with autofocus correction of the phase error for the same target

Stationary and moving target SAR images were used as inputs for the simulations in this study.

2. CNN Architecture

The CNN used in this study consisted of fifteen layers. The CNN has seven convolution layers each followed by a pooling layer and one fully connected layer. Each convolution layer consists of the convolution linear function, a batch normalization layer, and a ReLU activation function. The batch normalization layer normalizes the activations and gradients to speed up the network training and reduce the sensitivity to network initialization [13]. The convolution layer uses a 2 × 2 filter, a stride of one, and zero padding. The first convolution layer uses eight filters. The number of filters used in each subsequent convolution layer is doubled with the final convolution layer using 512 filters. Max pooling is used for each pooling layer. The pooling layers have a 2 × 2 output size and a stride of two. The fully connected layer has an output size of eight and it includes a

Softmax activation function and a classification layer. Figure 16 reveals a portion of the MATLAB code for the CNN.

```
convolution2dLayer(2,512,'Padding','same')
batchNormalizationLayer
reluLayer

maxPooling2dLayer(2,'Stride',2)

fullyConnectedLayer(8)
softmaxLayer
classificationLayer];
```

Figure 16. MATLAB view of part of CNN structure

3. Training and Testing

SAR images need to be separated into training and testing groups for each simulation. For each simulation, approximately 70% to 80% of the available images were used for training and 30% to 20% were used for testing. This ratio was kept to ensure the CNN had sufficient data to train on. The number of SAR images for each target type is shown in Table 2.

Table 2. Target types and number of SAR images for each

Target	2S1	BMP2	BTR70	M1	M2	M35	M60	T72
Images	879	502	504	729	724	729	874	503

C. SIMULATIONS

In this study, various simulations were used to test the accuracy of the CNN versus stationary and moving targets. These simulations were designed to help show the potential utility of CNNs for the purpose of ATR.

1. Stationary Targets

This first simulation used the stationary targets from the SAMPLE dataset to set a baseline accuracy for the CNN. The images selected for testing were chosen at random from the available images for each target. The number of images used for training and testing of the CNN for this simulation is shown in Table 3.

Table 3. Breakdown of the number of training and testing images for each target for the first simulation

Target	2S1	BMP2	BTR70	M1	M2	M35	M60	T72
Training	819	442	444	669	664	669	814	443
Testing	60	60	60	60	60	60	60	60

2. Slow Moving Targets

The second simulation used slow moving targets for the training and testing of the CNN. This simulation was designed to test if the CNN can still identify and learn the features of moving targets in SAR images. The testing images were selected by taking 30% to 20% of the images and all from the same elevation angle. The number of images used for training and testing of the CNN for this simulation is shown in Table 4.

Table 4. Breakdown of the number of training and testing images for each target for the second simulation

Target	2S1	BMP2	BTR70	M1	M2	M35	M60	T72
Training	594	402	404	555	551	554	589	403
Testing	285	100	100	174	173	175	285	100

3. Stationary Targets versus Moving Targets

The third simulation trained the CNN using the stationary SAR images and tested using moving SAR images. This simulation was designed to test if the CNN could still recognize and classify moving targets if the CNN is only trained on stationary targets. The number of images used for training and testing of the CNN for this simulation is shown in Table 5.

Table 5. Breakdown of the number of training and testing images for each target for the third simulation

Target	2S1	BMP2	BTR70	M1	M2	M35	M60	T72
Training	879	502	504	729	724	729	874	503
Testing	285	100	100	174	173	175	285	100

4. Slow Moving Targets versus Fast Moving Targets

This simulation used moving targets for both training and testing. The phase error applied to the testing images was greater than the phase error applied to the training images. The difference in phase error was used to simulate test targets that were moving faster than the targets used for training. Figure 17 shows the difference between the SAR images utilized for the slower moving targets and the faster moving targets. Figure 18 shows the difference in phase errors applied to the slower and faster moving targets. The number of images used for training and testing of the CNN for this simulation is shown in Table 6.

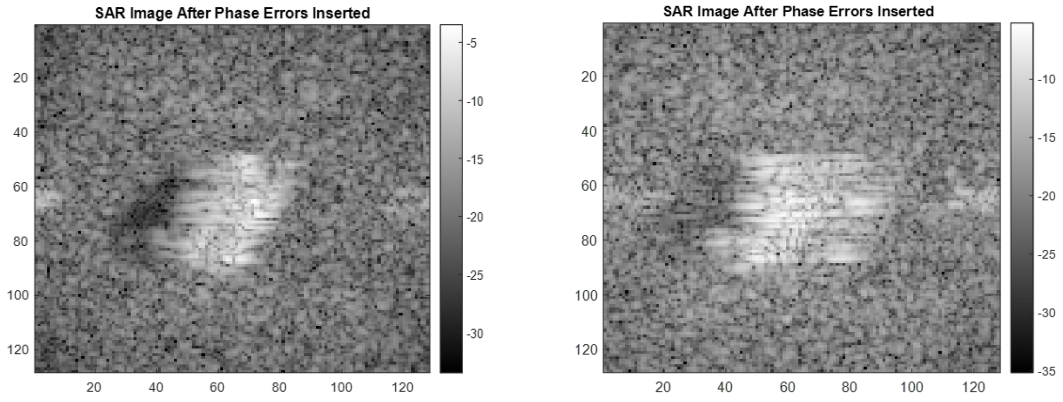

Figure 17. Example of slower target image on the left and faster target image on the right

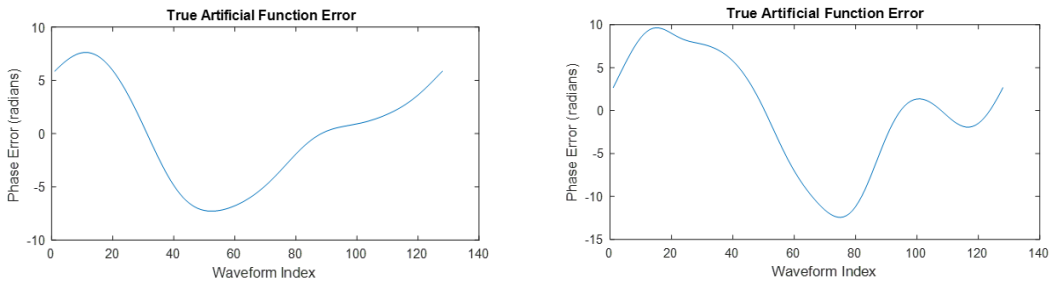

Figure 18. Example of phase error applied to slower target image on the left and faster target image on the right

Table 6. Breakdown of the number of training and testing images for each target for the fourth simulation

Target	2S1	BMP2	BTR70	M1	M2	M35	M60	T72
Training	594	402	404	555	551	554	589	403
Testing	285	100	100	174	173	175	285	100

This simulation was run to check if the CNN classification accuracy would change when a moving target was used for training rather than a stationary target.

5. Fast Moving Targets versus Slow Moving Targets

This simulation was similar to the previous simulation except the images used for training were the faster moving targets and the slower moving targets were used for testing. This simulation was run to see if training on images with larger phase errors could improve the accuracy of classifying images with smaller phase errors. The same number of images were used for training and testing as the previous simulation.

6. Stationary Targets versus Autofocused Targets

This simulation used the SAMPLE stationary SAR images for training and tested the CNN classification accuracy using images that had phase errors but were corrected using PGA code. This simulation tested the ability of autofocusing to correct phase errors

caused by moving targets in SAR imagery. The number of images used for training and testing of the CNN for this simulation is shown in Table 7.

Table 7. Breakdown of the number of training and testing images for each target for the sixth simulation

Target	2S1	BMP2	BTR70	M1	M2	M35	M60	T72
Training	879	502	504	729	724	729	874	503
Testing	285	100	100	174	173	175	285	100

THIS PAGE INTENTIONALLY LEFT BLANK

IV. RESULTS AND ANALYSIS

A. RESULTS

The results in this chapter are presented in the same order as they were discussed in the previous chapter.

1. Stationary Targets

This simulation, as discussed in the preceding chapter, used eight different stationary targets from the SAMPLE dataset to both train and test the recognition and classification capabilities of the CNN. Table 3 gives the number of target images used to train and test the CNN for this simulation. The average classification accuracy for this simulation was 99.65%, and the highest achieved accuracy was 99.79%. The classification accuracy results for this simulation are shown in Table 8. Figures 19 and 20 show the training progress and confusion matrix, respectively, for the second run of this simulation.

Table 8. Stationary targets results

Target	Run 1: Percent Correct	Run 2: Percent Correct	Run 3: Percent Correct	Average Percent Correct
2S1	98.33%	100%	100%	99.44%
BMP2	100%	100%	100%	100%
BTR70	100%	100%	100%	100%
M1	100%	100%	100%	100%
M2	96.67%	100%	98.33%	98.33%
M35	100%	100%	100%	100%
M60	100%	100%	100%	100%
T72	100%	98.33%	100%	99.44%
Average Percent Correct	99.38%	99.79%	99.79%	99.65%

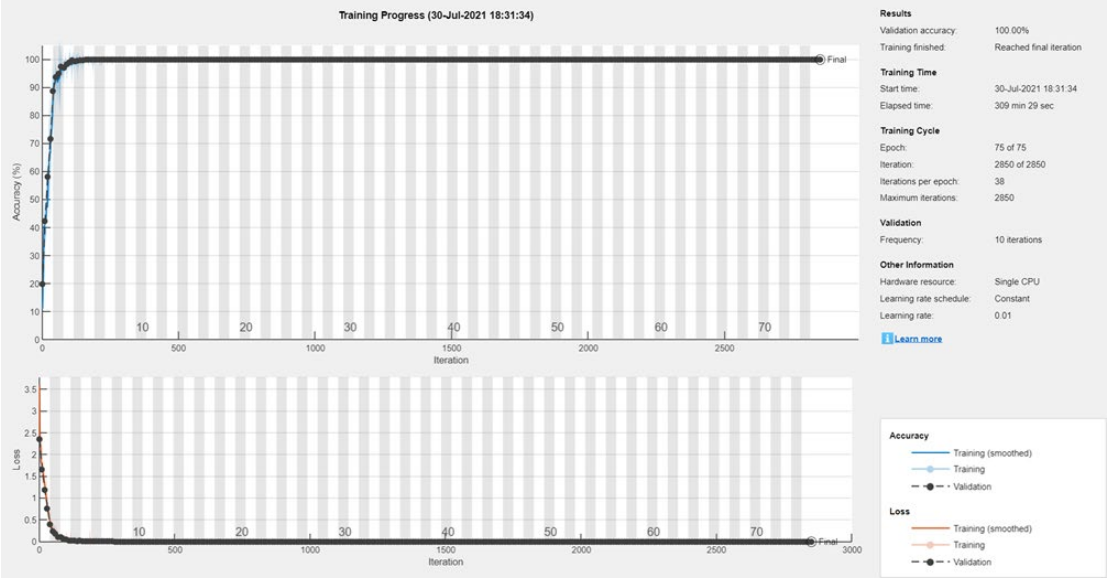

Figure 19. Training progress for stationary target simulation with 99.79% classification accuracy

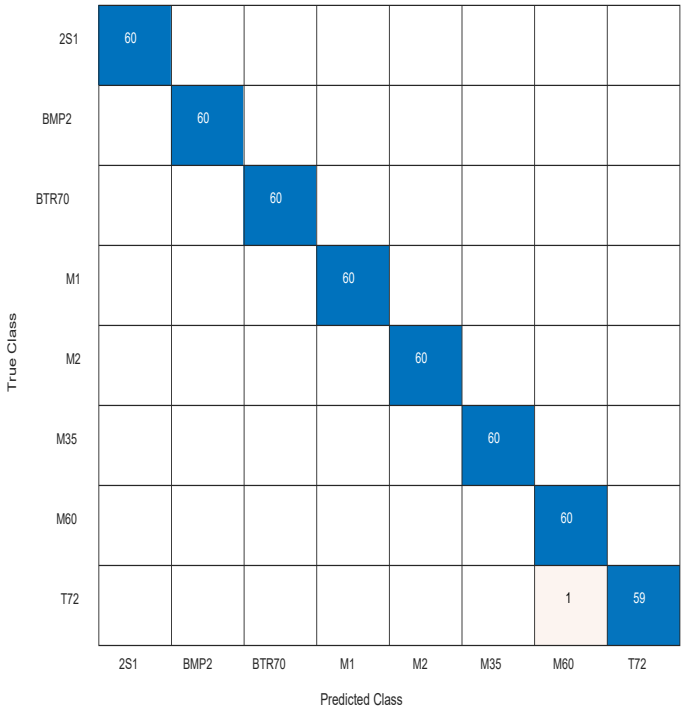

Figure 20. Confusion matrix for stationary target simulation with 99.79% classification accuracy

2. Moving Targets

This simulation used computer-generated moving target images for both training and testing. These images were generated as previously discussed by applying randomly generated phase errors to the SAMPLE SAR images. Table 4 gives the number of images used for training and testing of the CNN in this simulation. The average classification accuracy for this simulation was 86.30%, and the highest achieved accuracy was 86.78%. The classification accuracy results for this simulation are shown in Table 9. Figures 21 and 22 show the training progress and confusion matrix, respectively, for the first run of this simulation.

Table 9. Moving target results

Target	Run 1: Percent Correct	Run 2: Percent Correct	Run 3: Percent Correct	Average Percent Correct
2S1	96.14%	96.49%	97.19%	96.61%
BMP2	94%	89%	92%	91.67%
BTR70	96%	94%	91%	93.67%
M1	14.94%	17.24%	14.37%	15.52%
M2	98.27%	95.95%	95.38%	96.53%
M35	97.14%	96.57%	97.14%	96.95%
M60	99.30%	99.65%	99.30%	99.42%
T72	95%	91%	95%	93.67%
Average Percent Correct	86.78%	86.06%	86.06%	86.30%

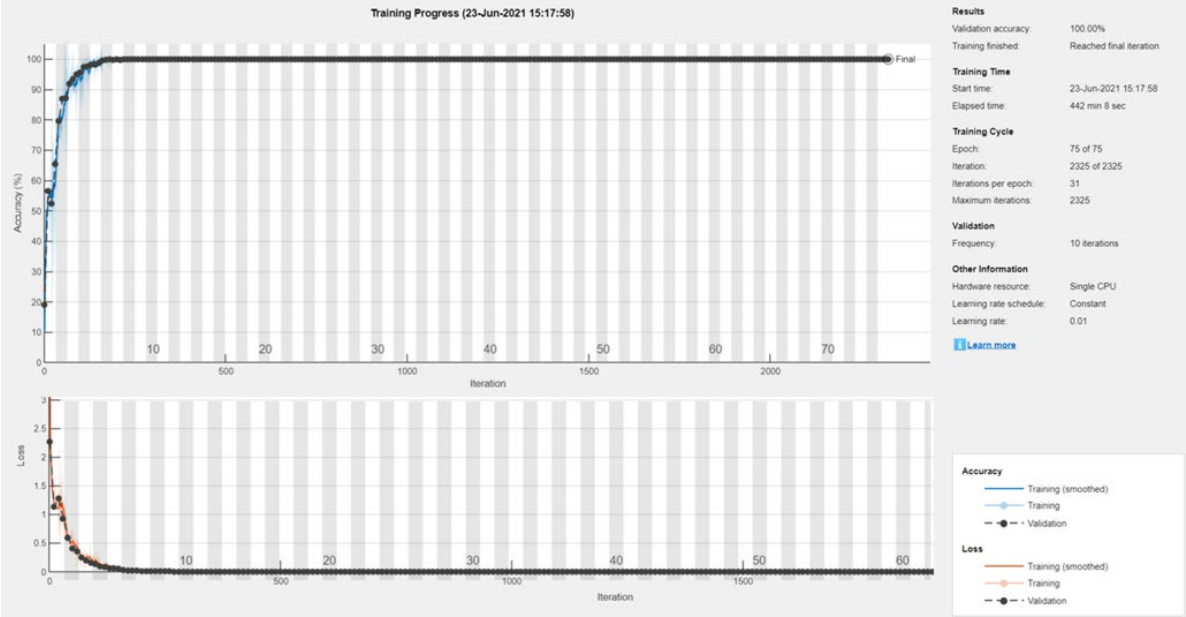

Figure 21. Training progress for moving target simulation with 86.78% classification accuracy

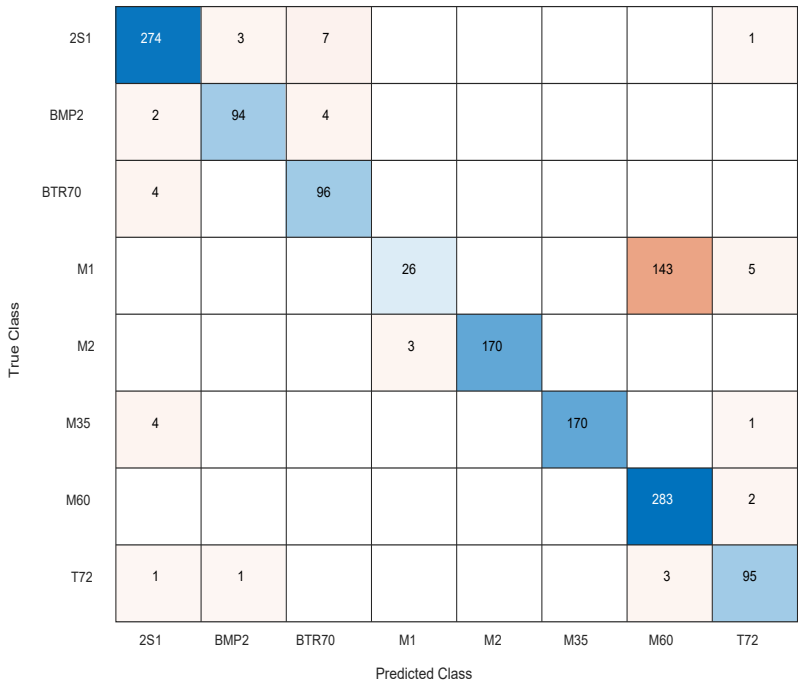

Figure 22. Confusion matrix for moving target simulation with 86.78% classification accuracy

3. Stationary Targets versus Moving Targets

The third simulation trained the CNN with stationary targets and used moving targets for testing. Table 5 gives the number of images used for training and testing the CNN in this simulation. The average classification accuracy for this simulation was 11.64%, and the highest achieved accuracy was 16.95%. The classification accuracy results for this simulation are shown in Table 10. Figures 23 and 24 show the training progress and confusion matrix, respectively, for the first run of this simulation.

Table 10. Stationary vs. moving targets results

Target	Run 1: Percent Correct	Run 2: Percent Correct	Run 3: Percent Correct	Average Percent Correct
2S1	0%	0%	0%	0%
BMP2	0%	0%	0%	0%
BTR70	13%	34%	1%	16%
M1	32.76%	74.14%	17.24%	41.38%
M2	0%	0%	0%	0%
M35	88.57%	19%	20.57%	40%
M60	0.00%	0.00%	0%	0.00%
T72	10%	0%	0%	3.33%
Average Percent Correct	16.95%	13.15%	4.81%	11.64%

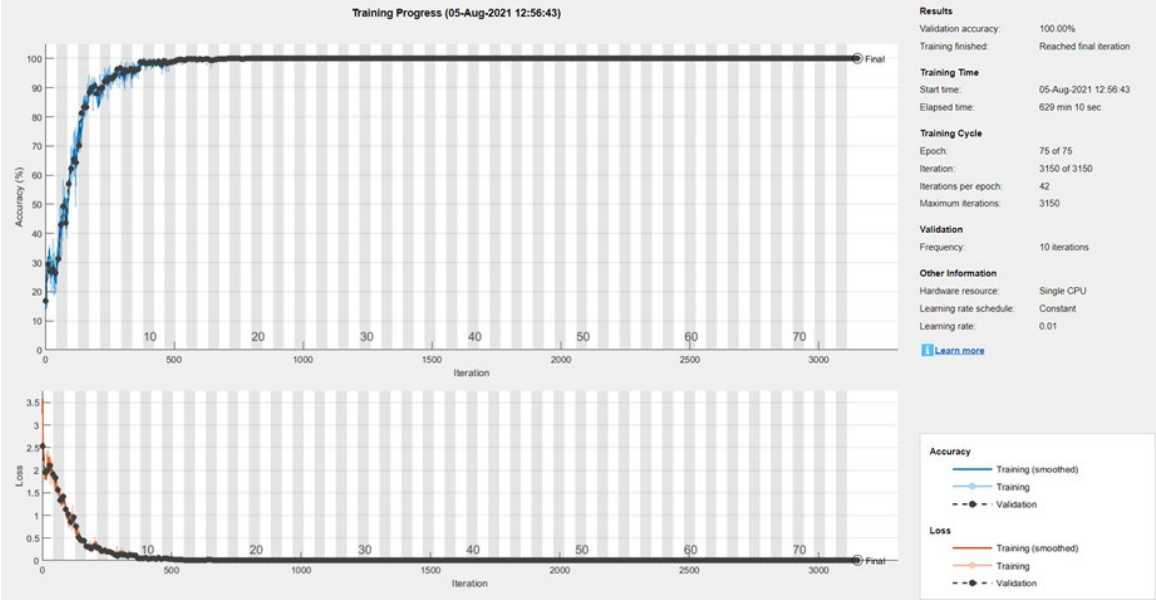

Figure 23. Training progress for moving target simulation with 16.95% classification accuracy

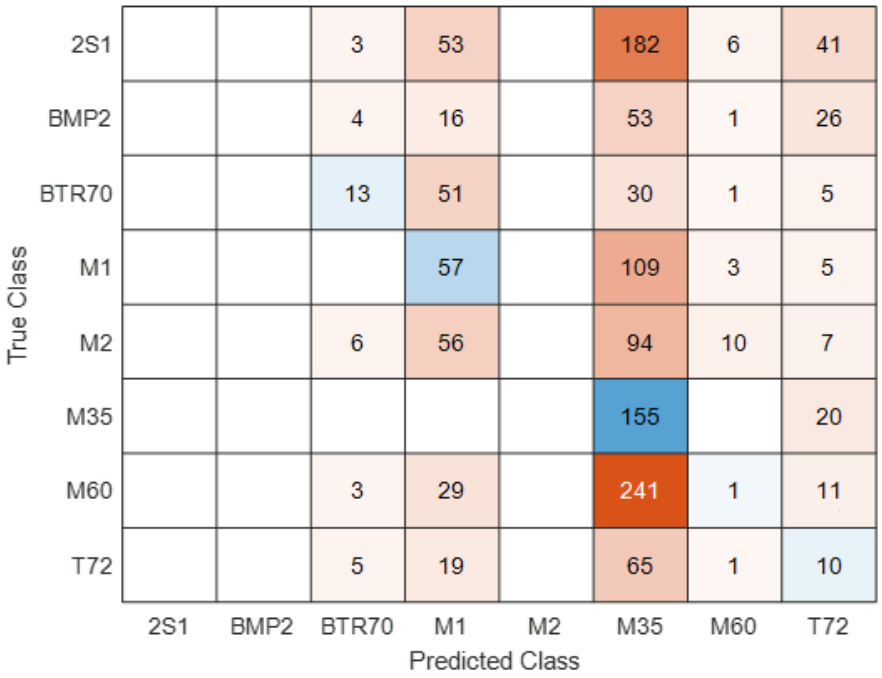

Figure 24. Confusion matrix for moving target simulation with 16.95% classification accuracy

4. Slow Moving Targets versus Fast Moving Targets

The fourth simulation used moving targets for both training and testing the CNN. The images used for training were images with smaller applied phase errors resembling comparatively slower moving targets. The images used for testing had larger applied phase errors to resemble faster moving targets. Table 6 gives the number of images used for training and testing. The average classification accuracy for this simulation was 61.49%, and the highest achieved accuracy was 63.36%. The classification accuracy results for this simulation are shown in Table 11. Figures 25 and 26 show the training progress and confusion matrix, respectively, for the first run of this simulation.

Table 11. Slow vs. fast moving targets results

Target	Run 1: Percent Correct	Run 2: Percent Correct	Run 3: Percent Correct	Average Percent Correct
2S1	65.61%	60%	64.56%	63.39%
BMP2	72%	55%	66%	64.33%
BTR70	85%	78%	78%	80.33%
M1	17.24%	9.20%	16.67%	14.37%
M2	0%	0%	0%	0%
M35	97.14%	96.57%	98.29%	97.33%
M60	84.91%	83.16%	83.86%	83.98%
T72	96%	98%	94%	96%
Average Percent Correct	63.36%	59.20%	61.93%	61.49%

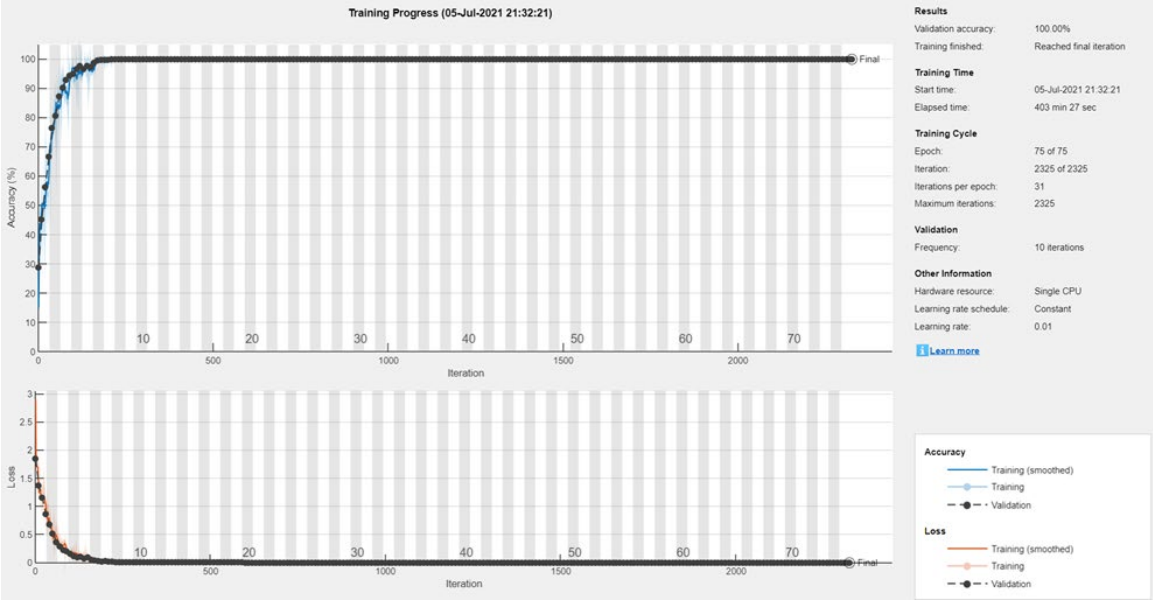

Figure 25. Training progress for moving target simulation with 63.36% classification accuracy

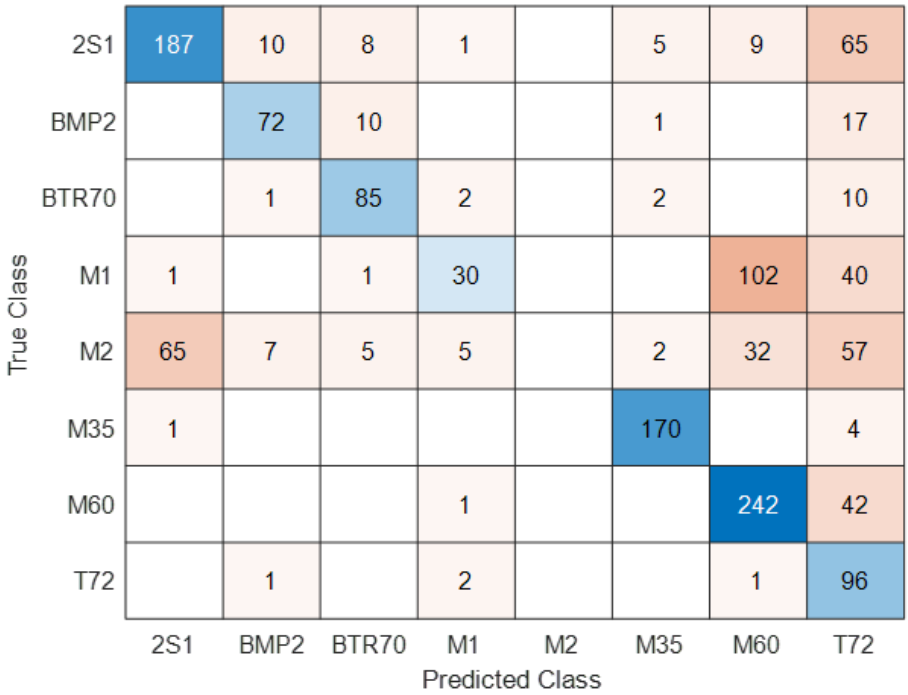

Figure 26. Confusion matrix for moving target simulation with 63.36% classification accuracy

5. Fast Moving Targets versus Slow Moving Targets

This simulation trained the CNN using the faster moving targets and tested using the slower moving targets. The average classification accuracy for this simulation was 90.95%, and the highest achieved accuracy was 91.81%. The classification accuracy results for this simulation are shown in Table 12. Figures 27 and 28 show the training progress and confusion matrix, respectively, for the second run of this simulation.

Table 12. Fast vs. slow moving targets results

Target	Run 1: Percent Correct	Run 2: Percent Correct	Run 3: Percent Correct	Average Percent Correct
2S1	96.49%	91.22%	94.74%	94.15%
BMP2	68%	67%	74%	69.67%
BTR70	90%	92%	90%	90.67%
M1	81.61%	86.21%	79.31%	82.38%
M2	97.11%	97.69%	97.11%	97.30%
M35	94.86%	97.71%	94.86%	95.81%
M60	99.30%	98.60%	99.65%	99.18%
T72	73%	88%	65%	75.33%
Average Percent Correct	90.88%	91.81%	90.16%	90.95%

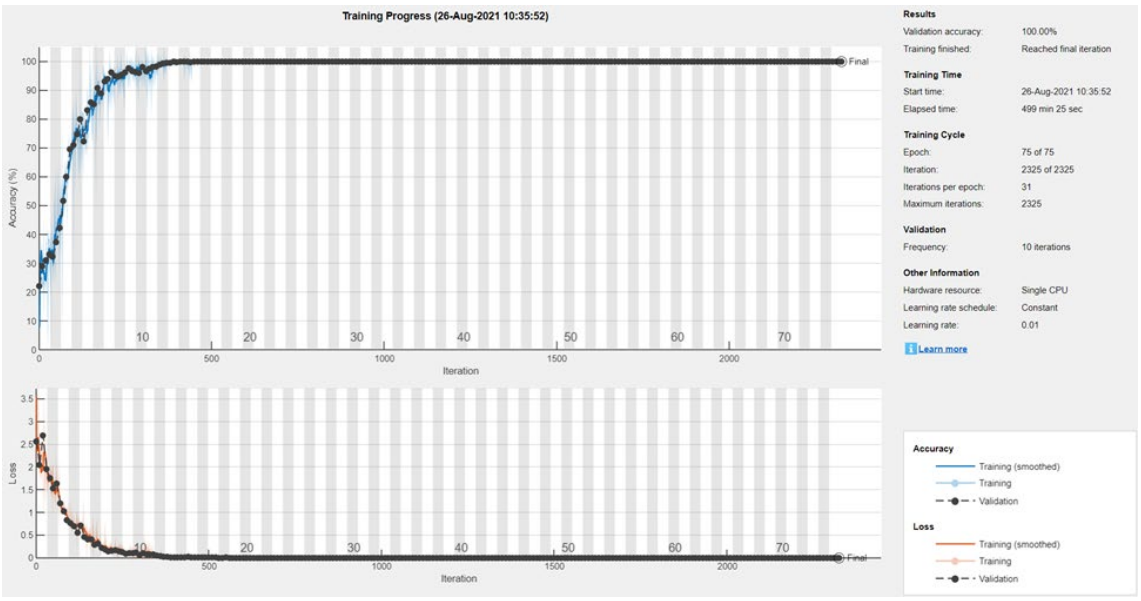

Figure 27. Training progress for moving target simulation with 91.81% classification accuracy

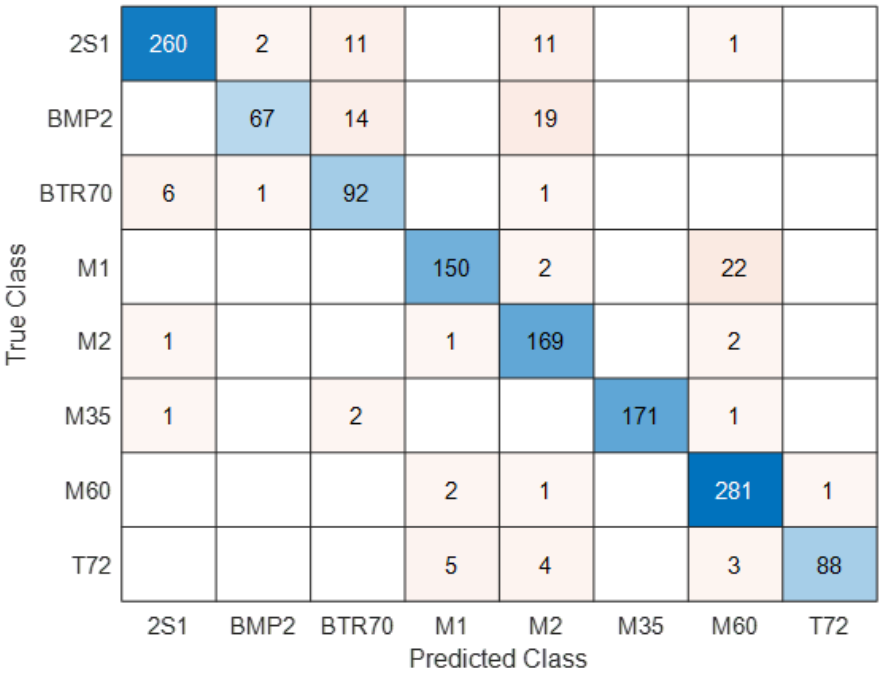

Figure 28. Confusion matrix for moving target simulation with 91.81% classification accuracy

6. Stationary Targets versus Autofocused Targets

The final simulation used stationary targets to train the CNN and autofocused images to test the CNN. Table 7 gives the number of training and testing images. The classification accuracy results for this simulation are shown in Table 13. Figures 29 and 30 show the training progress and confusion matrix, respectively, for the third run of this simulation.

Table 13. Stationary vs. autofocused targets results

Target	Run 1: Percent Correct	Run 2: Percent Correct	Run 3: Percent Correct	Average Percent Correct
2S1	0.06%	0.06%	0.08%	0.06%
BMP2	19%	19%	23%	20.33%
BTR70	84%	84%	80%	82.67%
M1	27.59%	27.59%	17.82%	24.33%
M2	0.08%	0.08%	20.81%	11.95%
M35	72%	72%	82.29%	75.43%
M60	0.08%	0.08%	30.18%	15.67%
T72	35%	35%	63%	44.33%
Average Percent Correct	26.22%	26.22%	34.91%	29.12%

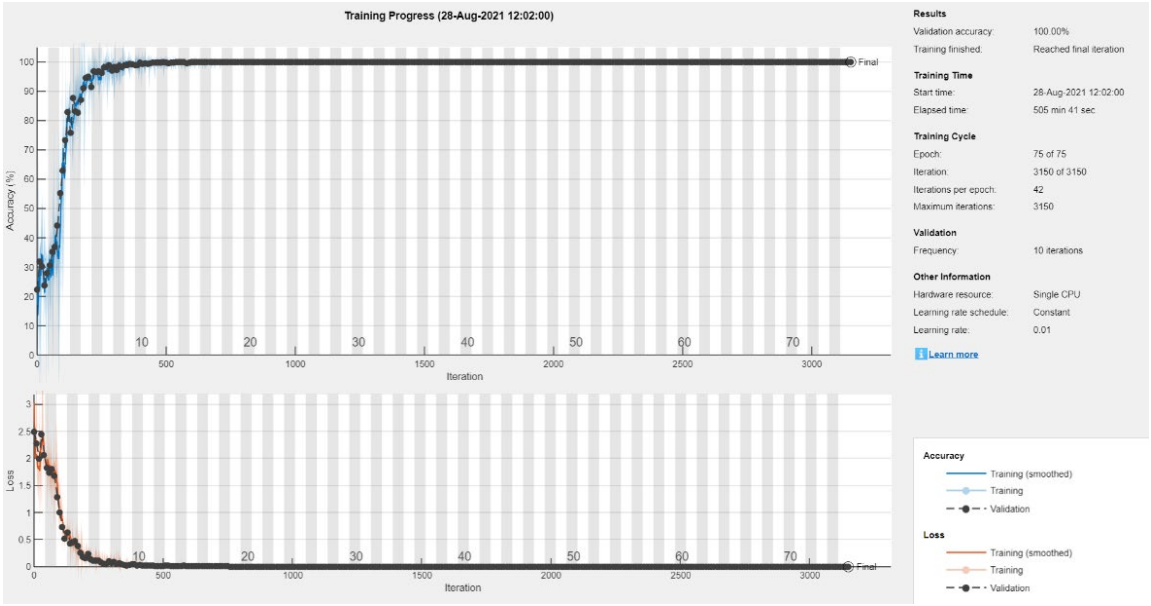

Figure 29. Training progress for autofocused target simulation with 34.91% classification accuracy

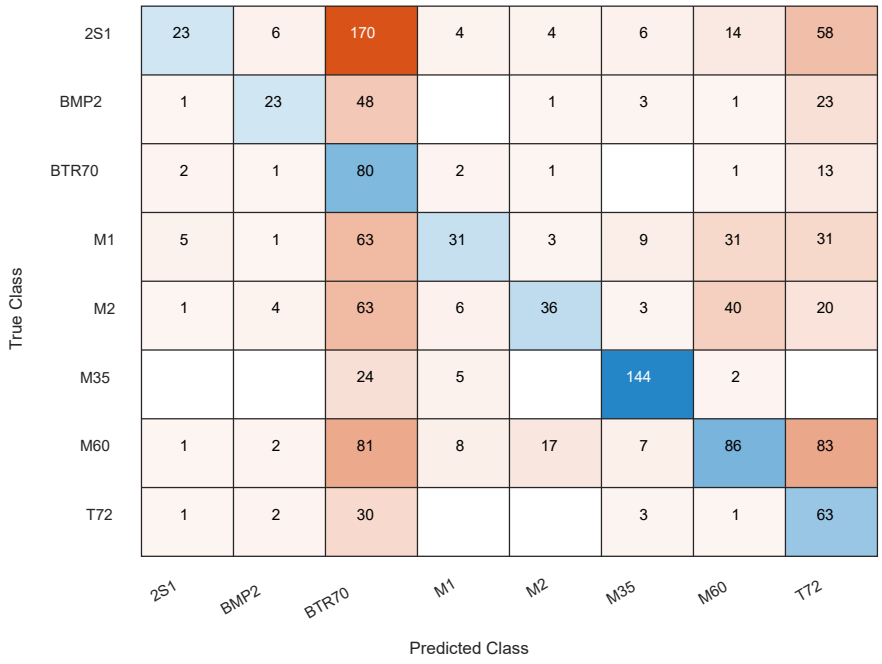

Figure 30. Confusion matrix for autofocused target simulation with 34.91% classification accuracy

B. ANALYSIS OF RESULTS

The first simulation in this study was designed to determine how accurately the utilized CNN could recognize and classify stationary targets. The average classification accuracy for this simulation was 99.65%. This simulation set the baseline achievable accuracy for the CNN while training and testing with the SAMPLE data. The goal of the other simulations was to determine if the CNN could possibly achieve the same accuracy with moving targets.

The second simulation in this study used moving targets for both training and testing. The purpose of this simulation was to compare the accuracy results to the simulation where stationary targets were used for training and moving targets were used for testing. This comparison was used to determine if training with moving targets would increase or decrease CNN accuracy when classifying other moving targets. The average classification accuracy for this simulation was 86.30%. The CNN was not able to achieve as high of accuracy as the first simulation, with the moving targets decreasing the accuracy of the CNN by approximately 13%.

The third simulation utilized stationary targets to train the CNN and moving targets to test it. This simulation achieved an average classification accuracy of 11.64%, which is approximately 75% less accurate on average than the CNN trained with moving target imagery. The results of the second and third simulations show that the CNN is able to learn characteristics of moving targets even when the motion is not uniform between each target image.

The fourth simulation trained the CNN with moving target images designed to resemble relatively slow-moving targets. The CNN was tested using target images designed to resemble faster moving targets. The purpose of this simulation was to determine if testing the CNN using faster moving target images, meaning images with more smearing in comparison to the training images, would impact the accuracy of the CNN. The average accuracy of this simulation was 61.49%. Training the CNN with slower moving targets than those it was tested on decreased the CNN accuracy by approximately 25%. Training the CNN with slower moving targets was less effective than training with

targets of roughly similar speeds, but it was still far more effective than training with stationary targets by approximately 50%.

The training and testing data used for the fifth simulation was the opposite of the data used for the fourth simulation. The CNN was trained using the faster moving target images and tested using the slower target images. The average accuracy for the fifth simulation was 90.59%. Training the CNN with the faster moving targets improved the accuracy of the CNN by 4.29% in comparison to the second simulation, 29.1% in comparison to the fourth simulation, and 78.95% in comparison to the third simulation.

The sixth and final simulation used stationary target images for training and autofocused images for testing. The purpose of this simulation was to determine if autofocus is an effective method of correcting the phase error caused smearing in moving target SAR images. The average accuracy for this simulation was 29.12%. The average accuracy for this simulation was 17.48% higher than the accuracy of the third simulation which did not use autofocus. Although the autofocus improved the CNN classification accuracy it was still much lower than the accuracy of the CNN trained using moving target images.

Overall using the faster moving target images to train the CNN achieved the highest accuracy for classifying moving targets and was the closest method to achieving the same accuracy levels as the CNN that classified stationary targets. The CNN that was trained using the faster moving targets achieved a classification accuracy of approximately 9% lower than the CNN that was trained and tested on the stationary target images. Table 14 summarizes the average classification accuracy of each simulation.

Table 14. Average classification accuracy of each simulation

Simulation	Average Accuracy
1: Stationary targets	99.65%
2: Moving targets	86.30%
3: Stationary target training vs. moving target testing	11.64%
4: Slow target training vs. fast target testing	61.49%
5: Fast target training vs. slow target testing	90.59%
6: Stationary target training vs. autofocused moving target testing	29.12%

V. CONCLUSION AND FUTURE WORK

A. CONCLUSION

The military operating environment has become increasingly characterized by the use of sensors to rapidly obtain vast amounts of information. Advanced sensor systems that are used today can rapidly provide critical intelligence on adversary units such as their size, movements, and actions. The more reliable information provided to military commanders, the more effectively they can make crucial decisions needed for mission success. Rapid collection and synthesis of various forms of accurate intelligence information allows military commanders to make decisions and act quickly. The ability for commanders to decide and act more swiftly than their adversaries provides a distinct advantage. SAR is a sensor that, combined with ATR image processing software such as a CNN, can provide valuable intelligence.

This study aimed to determine if a CNN could be used to reasonably identify moving targets in SAR imagery on top of its already proven capability of identifying stationary targets. As previously stated, ATR provides advantageous information to a military commander, but ATR of specifically moving targets can provide especially insightful information. Recognizing moving targets can provide commanders with awareness into the activity and intentions of adversary forces as well as early warning of the movements of high value targets.

Several experiments were conducted to analyze possible methods for accurate ATR of moving targets. One of the methods utilized achieved a 90.59% accuracy for identifying moving targets, which was approximately 9% less accurate than the accuracy of the CNN for identifying stationary targets. This high level of accuracy was achieved by training the CNN with targets with a larger average phase error than the images its was asked to classify. This was slightly unexpected as it was originally predicted that autofocus used to correct image phase errors would provide the highest ATR accuracy.

While conducting this study, a few problems were observed in relation to autofocus that likely prevented the accuracy from being higher for that specific simulation. The center

shift of the target image data can cause issues when the data gets converted back into the corrected image. In the autofocused images, the targets are frequently no longer centered in the image chips. In extreme cases, the target appears shifted to one of the image edges with some of the target data being cut off and wrapped to the opposite image edge. Figure 31 shows an example of this extreme case.

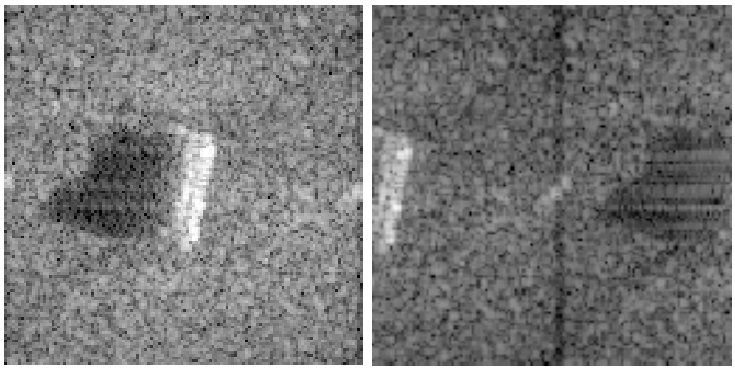

Figure 31. Example of stationary target image on the left and example of the same target after autofocusing on the right

In the right image of Figure 31, there is also an observable dark vertical line slightly to the right of the center of the image. This line shows up in every autofocused image. It appears that this darker image data originally existed on the edges of the SAMPLE images, but it gets shifted more into the center of the images when applying autofocusing. These differences between the autofocused images and the stationary images are likely the cause of the low accuracy of the CNN when tested on the autofocused images.

In conclusion, this study looked at ATR of stationary and moving targets in SAR imagery using a CNN and the effectiveness of various methods of achieving this task. In this study, the CNN achieved a 99.65% average accuracy for classifying stationary targets and a 90.59% accuracy for classifying moving targets.

B. FUTURE WORK

In this thesis, the best method for classifying moving targets was determined to be training the CNN with data also comprised of moving target images. For future research,

more moving target images could be created from the SAMPLE data set for the purpose of training the CNN. A database could be created out of target images with a broader range of phase errors applied to the target images. In this study, a slightly larger range of phase errors increased the CNN accuracy, this increase could possibly be amplified with an even larger range of phase errors. Another possibility for future research is to develop a CNN that only identifies whether target images show moving or stationary targets. These target images could then be separated into moving or stationary target bins and then classified by one of two different CNNs that have been trained on either moving or stationary targets. Lastly, this study was done using SAR images that were centered on the targets to be identified without any other objects in the images. To make the study more realistic, a total scene SAR image could be created involving multiple targets as well as other objects such as buildings. A machine learning algorithm could be developed to pick out and identify targets from within the image.

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

[1] M. A. Richards, J. A. Scheer, and W. A. Holm, *Principles of Modern Radar*, vol. 1: Basic Principles. Raleigh, NC, USA: SciTech Publishing, 2010.

[2] B. Lewis, T. Scarnati, E. Sudkamp, J. Nehrbass, S. Rosencrantz, and E. Zelnio, "A SAR dataset for ATR development: The Synthetic and Measured Paired Labeled Experiment (SAMPLE)," in *Algorithms for Synthetic Aperture Radar Imagery XXVI*, May 2019, vol. 10987, p. 109870H. [Online]. Available: doi: 10.1117/12.2523460.

[3] B. Lewis, T. Scarnati, E. Sudkamp, J. Nehrbass, S. Rosencrantz, and E. Zelnio, "SAMPLE-Synthetic And Measured Paired Labelled Experiment - ATRPEDIA DIST C." [Online]. Available: https://restricted.vdl.afrl.af.mil/programs/atrpedia/dist_c/wiki/SAMPLE-Synthetic_And_Measured_Paired_Labelled_Experiment (accessed Sep. 20, 2021).

[4] The Mathworks, Inc. 2021. *MATLAB, ver. 2021a*. [Online]. Available: www.mathworks.com

[5] J. C. Curlander and R. N. McDonough, *Synthetic Aperture Radar: Systems and Signal Processing*. New York, NY, USA: John Wiley & Sons, Inc., 1991.

[6] A. Moreira, P. Prats-Iraola, M. Younis, G. Krieger, I. Hajnsek, and K. P. Papathanassiou, "A tutorial on synthetic aperture radar," *IEEE Geosci. Remote Sens. Mag.*, vol. 1, no. 1, pp. 6–43, Mar. 2013, [Online]. Available: doi: 10.1109/MGRS.2013.2248301.

[7] C. Ozdemir, *Inverse Synthetic Aperture Radar Imaging with MATLAB Algorithms*. Hoboken, NJ, USA: John Wiley & Sons, Incorporated, 2012. Accessed: Jul. 12, 2021. [Online]. Available: http://ebookcentral.proquest.com/lib/ebook-nps/detail.action?docID=818515

[8] C. V. Jakowatz, D. E. Wahl, P. H. Eichel, D. C. Ghiglia, and P. A. Thompson, "Spotlight-mode synthetic aperture radar: A signal processing approach," Norwell, MA, USA: Kluwer Academic Publishers, 1996, pp. 221–271.

[9] R. Yamashita, M. Nishio, R. K. Gian Do, and K. Togashi, "Convolutional neural networks: An overview and application in radiology," *Insights Imaging*, Jun. 2018, [Online]. Available: https://doi.org/10.1007/s13244-018-0639-9

[10] U. K. Majumder, E. P. Blasch, and D. A. Garren, *Deep Learning for Radar and Communications Automatic Target Recognition*. Norwood, MA, USA: Artech House, 2020.

[11] K. O'Shea and R. Nash, "An introduction to convolutional neural networks," *ArXiv151108458 Cs*, Dec. 2015, Accessed: Jul. 18, 2021. [Online]. Available: http://arxiv.org/abs/1511.08458

[12] "Learn about convolutional neural networks - MATLAB & Simulink." [Online]. Available: https://www.mathworks.com/help/deeplearning/ug/introduction-to-convolutional-neural-networks.html (accessed Aug. 02, 2021).

[13] "Create simple deep learning network for classification - MATLAB & Simulink example." [Online]. Available: https://www.mathworks.com/help/deeplearning/ug/create-simple-deep-learning-network-for-classification.html (accessed Aug. 03, 2021).

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California